Package ‘SynthpopDecoinceur’

September 10, 2018

Type Package

Title X

Version 1.0

Date 2018-09-10

Author D. Bonnery

Maintainer D. Bonnery <dbonnery@umd.edu>

Description Data

Remotes tidyverse/magrittr

Depends Synthpop

License GPL (>= 2)

LazyLoad yes

LazyData true

R topics documented:

exportRpackagedata_to_sas7bdatlibrary

export all tables from an installed R data package to sas7bdat data files in a given library

Description

export all tables from an installed R data package to sas7bdat data files in a given library

Usage

```
exportRpackagedata_to_sas7bdatlibrary(package, path_to_export_to)
```

Arguments

package	package to be transformed
SAS_library_path	path to export the data

Value

nothing

Examples

```
exportRpackagedata_to_sas7bdatlibrary("datasets")
```

GeneralReversetransposefunction

General Transpose function

Description

General Transpose function

Usage

```
GeneralReversetransposefunction(TtableA, key)
```

Arguments

key	A list of variables (columns of the transposed table)
table	A dataframe

Value

A list: first element of the list is a dataframe, the transposed version of the orioginal table. Second element is a key to allow back transposition

Examples

```
N<-100;tableA<-data.frame(id1a=as.factor(sample(30,N, replace=TRUE)),
id1b=as.factor(sample(4,N, replace=TRUE)),
id2a=sample(letters[1:3],N,replace=TRUE),
id2b=sample(letters[1:4],N,replace=TRUE),
cont1=rchisq(N,1),
cont2=rchisq(N,1),
factor1=as.factor(sample(letters[1:3],N,replace=TRUE)),
factor2=as.factor(sample(letters[1:4],N,replace=TRUE)),
char1=sample(letters[1:3],N,replace=TRUE),
char2=sample(letters[1:4],N,replace=TRUE),stringsAsFactors=FALSE)
#Add missing
tableA[5:10]<-lapply(tableA[5:10], function(x){x[sample(N,10)]<-NA;x})
id1=c("id1a","id1b")
id2=c("id2a","id2b")
toto<-Generaltransposefunction(tableA,id1,id2)
TtableA=toto$TtableA;key=toto$key
RtableA=GeneralReversetransposefunction(TtableA,key)
identical(tableA,RtableA)
```

Generaltransposefunction *General Transpose function*

Description

General Transpose function

Usage

```
Generaltransposefunction(tableA, id1, id2)
```

Arguments

id1	A list of variables (rows)
id2	A list of variables (columns of the transposed table)
table	A dataframe

Value

A list: first element of the list is a dataframe, the transposed version of the orioginal table. Second element is a key to allow back transposition

Examples

```
N<-100;tableA<-data.frame(id1a=as.factor(sample(30,N, replace=TRUE)),
id1b=as.factor(sample(4,N, replace=TRUE)),
id2a=sample(letters[1:3],N,replace=TRUE),
id2b=sample(letters[1:4],N,replace=TRUE),
cont1=rchisq(N,1),
cont2=rchisq(N,1),
factor1=as.factor(sample(letters[1:3],N,replace=TRUE)),
factor2=as.factor(sample(letters[1:4],N,replace=TRUE)),
char1=sample(letters[1:3],N,replace=TRUE),
char2=sample(letters[1:4],N,replace=TRUE),stringsAsFactors=FALSE)
#Add missing
tableA[5:10]<-lapply(tableA[5:10], function(x){x[sample(N,10)]<-NA;x})
id1=c("id1a","id1b")
id2=c("id2a","id2b")
toto<-Generaltransposefunction(tableA,id1,id2)
```

Generaltransposefunctionsimple
Simple General Transpose function

Description

Simple General Transpose function

Usage

```
Generaltransposefunctionsimple(tableA, id1, id2)
```

Arguments

tableA	A dataframe
id1	A list of variables (rows)
id2	A list of variables (columns of the transposed table)

Value

A data frame

Examples

```
N<-100;tableA<-data.frame(id1a=as.factor(sample(30,N, replace=TRUE)),
id1b=as.factor(sample(4,N, replace=TRUE)),
id2a=sample(letters[1:3],N,replace=TRUE),
id2b=sample(letters[1:4],N,replace=TRUE),
cont1=rchisq(N,1),
cont2=rchisq(N,1),
factor1=as.factor(sample(letters[1:3],N,replace=TRUE)),
factor2=as.factor(sample(letters[1:4],N,replace=TRUE)),
char1=sample(letters[1:3],N,replace=TRUE),
char2=sample(letters[1:4],N,replace=TRUE),stringsAsFactors=FALSE)
#Add missing
tableA[5:10]<-lapply(tableA[5:10], function(x){x[sample(N,10)]<-NA;x})
id1=c("id1a","id1b")
id2=c("id2a","id2b")
toto<-Generaltransposefunctionsimple(tableA,id1,id2)
```

ggplot_missing *Create missing chart*

Description

Create missing chart

Usage

```
ggplot_missing(x, reordonne = FALSE)
```

Arguments

x	a dataframe
reordonne	a boolean

Value

a ggplot graph

Examples

```
library(reshape2)
library(ggplot2)
library(plyr)
library(magrittr)
X=cars
for(i in 1:40){
  X[sample(1:50,1,replace=TRUE),sample(1:2,1,replace=TRUE)]<-NA}
ggplot_missing(X,reordonne=TRUE)
ggplot_missing(X,reordonne=FALSE)createallautomaticRMD(schema="SDP")
```

ggplot_missing2 *Create missing chart*

Description

Create missing chart

Usage

```
ggplot_missing2(X, reordonne = TRUE, keep = NULL)
```

Arguments

X	a dataframe
reordonne	a boolean
keep	a boolean

Value

a ggplot graph

Examples

```
library(reshape2)
library(ggplot2)
library(plyr)
X=cars
X$year=sample(2012:2017,nrow(cars),replace=TRUE)
for(i in 1:40){
 X[sample(1:50,1,replace=TRUE),sample(1:2,1,replace=TRUE)]<-NA}
ggplot_missing2(X,keep="year")
```

missing.summary *tablename="TRAIT_LKP" automaticdataf(tablename)*

Description

tablename="TRAIT_LKP" automaticdataf(tablename)

Usage

```
missing.summary(X, info2 = NULL)
```

Index